Using the Arcane Mirage Template Project

Step 1: Download and Install Unreal Engine

- If you haven't already, download and install Unreal Engine from the official Unreal Engine website (https://www.unrealengine.com/en-US/download/).
- Follow the installation instructions for your specific platform (Windows, macOS, or Linux).

Step 2: Download the Project

- You can find our Template Project **here!**

Step 3: Extract the Project Files

- Locate the downloaded template file (**e.g PixelStreamingTemplate.zip**) and extract its contents anywhere on your computer (for example, C:\MyUnrealProjects\AMTemplate).

Step 4: Open Unreal Engine

- Launch the Unreal Engine Editor by running the "Unreal Engine Launcher" or the "Unreal Editor" application, depending on your installation. You can either open the template project or import its files into a project of your own.

Step 5 Option A: Open the Template Directly

- In the Unreal Engine Editor, click on "File" and select "Open Project."
- Browse to the folder where you extracted the Pixel Streaming Project Template files.
- Select the project file (usually with a .uproject extension) for the template and click "Open."

Step 5 Option B: Import the Template into an Existing Project

- Open your existing Unreal Engine project in the Unreal Engine Editor.
- In the Content Browser of your existing project, right-click on the folder where you want to import the Pixel Streaming Project Template components.
- Select "Import."
- Navigate to the folder where you extracted the Pixel Streaming Project Template files.
- Select the contents of the template directory (including the Blueprint files, assets, and other resources) and click "Import."

Step 6: Customize and Modify

- Explore the HUD Blueprint, first person character and freelook pawn,, UI widgets, and other components within your project.
- Customize and modify these components as needed to fit your project's specific requirements. You can do this by opening the associated Blueprints and UI widgets, and then editing their properties and scripts.

Step 7: Build and Test

- After making your customizations, build and test your project within the Unreal Engine Editor to ensure everything works as intended.
- Use the Play button in the editor to test your project in various modes and with different settings.

Step 8: Upload to the Arcane Mirage Platform

- Sign in to your Arcane Mirage account (assuming you have one). If you don't, you will need to create an account.
- Once logged in, you can find all the necessary information to upload your project here:
  https://docs.arcanemirage.com/arcane-mirage-platform/3.-uploading-your-first-project

Arcane Mirage Template Project Documentation

Introduction

Welcome to the Arcane Mirage Pixel Streaming Template Project. This template is designed to help you quickly set up a pixel streaming project with the following features:

First-Person Character Controller: A character controller capable of freely walking and zooming in the camera out of the box!
Free-Look Character Controller: A character controller that enables free movement around the environment, no strings to hold you down.
Mobile Controls: A simple device agnostic mobile interface that will allow you to test your project on any platform.
Simple UI: A user interface widget with buttons for switching between first-person and free-look character controllers, replaying your very own sequences, activating and deactivating mobile controls, and opening the options menu.
Options Menu: An options menu that allows users to customize the project's graphic quality, adjust mouse sensitivity, and control the camera's field of view, all of this on the fly!

In this documentation, we will explain each component and how users can interact with and modify them as needed.

HUD Blueprint (BP_AMHUD)

Overview

The HUD Blueprint is responsible for managing the user interface and handling communications between various in-game functions. Most of the UI elements call events from the BP_AMHUD blueprint in order to carry out the specified action.

Key Functions:

Each of the functions carried out in the BP_AMHUD blueprint has a simple comment explaining what it does over each event or group of nodes carrying out a function.

Freelook Toggle: Sets the Free Look pawn actor's position to the current position of the character controller and then possesses it, enabling the user to control it.

Return First Person: Returns the user to the first person character wherever it was before the freelook toggle was used.

Toggle Touch Interface: This event switches between the default AM_VirtualJoystick and none to enable or disable touch controls for mobile devices. The condition to toggle between these two is given by the mobile controls button in the WBP_AMUI widget blueprint.

Play Sequence: This event checks that the variable "My Sequence Player" isn't empty and then if there is a sequence to play it starts playing.

HideUI/ShowUI: Sets the visibility of the UI to hidden/Visible.

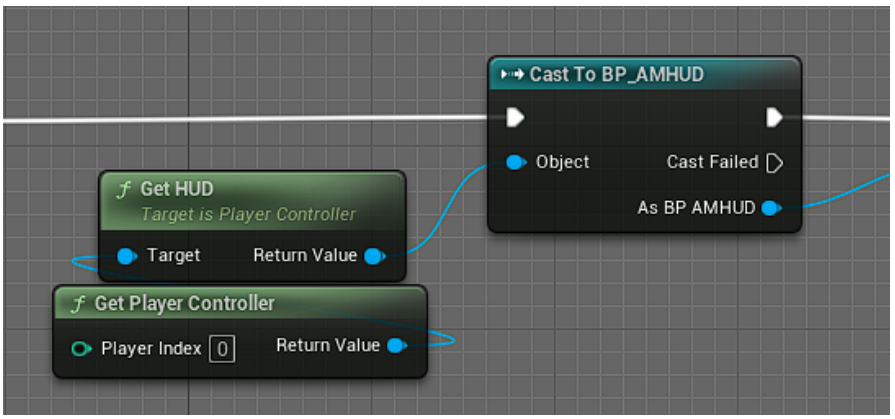
Customization

To modify the HUD Blueprint to suit your project's needs, follow these steps:

- Open the HUD Blueprint (BP_AMHUD) in the Unreal Engine Editor.
- Go to "Event Graph" to visualize the blueprints and here you can freely edit any of the provided events or even add your own!.
- An example for an event would be **Event BeginPlay:** the "Begin Play" event is a fundamental event that is commonly used in Blueprints to execute actions or behaviors when an actor or object is first initialized and begins to exist in the game world. It is often used for setup tasks, initialization, or any actions that need to occur at the start of an actor's life cycle. In the case of the HUD it will be called when the project starts running. In this particular blueprint the following happens in this event:
- Loads UI on Start-Up and creates a reference to it.
- Sets a reference to the first person character.
- Sets a reference to the free look pawn
- Sets a reference to a sequence for video replays in the MySequencePlayer variable, this is initially set to none and you're free to change it to any sequence you create yourself.

  Swap Character Controllers: Allows the user to switch between first-person and free-look character controllers.

- All of the events in the HUD can be called from anywhere else by adding the following snippet:



First-Person Character (BP_AMFirstPersonCharacter)

Overview

The first-person character controller provides a basic player character with the ability to walk and zoom the camera in.

Key Functions:

Walking: The character can move forward, backward, left, and right using standard movement controls.
Zoom In: The user can zoom in to get a closer look at objects or details in the world with the Z key.

Customization

To customize the first-person character controller:

- Open the First-Person Character blueprint (BP_AMFirstPersonCharacter) in the Unreal Engine Editor.
- Go to “Event Graph” to visualize the blueprints and here you can freely edit any of the provided events or even add your own!.

Inside this event graph you will find the following events:

Beginplay: As explained previously, Begin Play runs when the game object is first loaded in the case of the first person character it does the following:

- Sets the visibility of the mouse.

- Loads the control mappings from Unreal’s enhanced input system.
- Sets the input mode and focus to the viewport.

Zoom toggle: This is bound to the Z key and allows the camera to zoom in and out as a toggle.

Camera/Movement Input: These events and the code within their respective blocks use Unreal’s enhanced input system to handle character and camera movements.

Event Tick: The Tick event is triggered once per frame, typically 30 to 60 times per second, depending on the frame rate settings and hardware capabilities. This makes it suitable for handling continuous and real-time game logic.

In this particular event, we are handling the Zoom event, checking wether the condition for zoom is actually enabled and then activating the zoom as an example of what you can do on “Tick”.

Free-Look Character (BP_AMFreeLookPawn)

Overview

The free-look character controller enables the user to move freely around the game environment, offering a different perspective from the first-person view.

Key Functions:

Free Movement: The character can move in any direction, offering unrestricted exploration of the environment.

Customization

To customize the Free-Look character controller:

- Open the First-Person Character blueprint (BP_AMFirstPersonCharacter) in the Unreal Engine Editor.
- Modify the character's appearance, abilities, or behavior to suit your project's requirements.
- Go to “Event Graph” to visualize the blueprints and here you can freely edit any of the provided events or even add your own!.

Inside this event graph you will find the following events:

Beginplay: As explained previously, Begin Play runs when the game object is first loaded in the case of the first person character it does the following:

- Sets the visibility of the mouse.
- Loads the control mappings from Unreal's enhanced input system.
- Sets the input mode and focus to the viewport.

Camera Input: These events and the code within their respective blocks use Unreal's enhanced input system to handle character and camera movements.

Simple UI Widget (WBP_AMUI)

Overview

The simple UI widget provides buttons that allow users to switch between character controllers, replay sequences, and open the options menu.

Key Functions:

Switch Character Controllers: Toggles between the first-person and free-look character controllers.
Replay Sequence: Initiates the replay of a sequence assigned in the HUD.
Open Options Menu: Displays the options menu for adjusting graphics quality, mouse sensitivity, and field of view.

Customization

To customize the UI widget:

- Open the widget blueprint (WBP_AMUI) in the Unreal Engine Editor.
- Modify the widget's appearance or add additional buttons and functionality.
- Adjust the button behaviors by editing the associated Blueprint scripts in the event graph.

Inside this event graph you will find events that carry out each of the previously mentioned key functions by calling the respective events from the HUD (**BP_AMHUD**) whenever their respective buttons are clicked (On Clicked events).

The "On Clicked" event for the options menu instantiates the options menu UI widget and then adds it to the viewport.

Options Menu UI Widget (WBP_AMOptions)

Overview

The options menu UI widget allows users to customize the project's graphical settings, mouse sensitivity, and camera field of view.

Key Functions:

Graphic Quality: Allows users to select different graphics quality settings for the game.
Mouse Sensitivity: Adjusts the sensitivity of the mouse or input device.
Field of View: Controls the camera's field of view, affecting the player's perspective.

Customization

To customize the options menu UI widget:

- Open the widget blueprint (WBP_AMOptions) in the Unreal Engine Editor.
- Modify the widget's appearance or add additional buttons and functionality.
- Adjust the button behaviors by editing the associated Blueprint scripts in the event graph.

Inside this event graph you will find the following events:

Event Construct: The "Event Construct" is essentially the initialization event for a widget, similar to the “Begin Play” event in a regular blueprint. When the widget is created and added to the viewport or parent widget, the "Event Construct" event is triggered once.

- In the case of this blueprint, the following happens in this event:
- Sets a reference to the graphics scalability settings.
- Sets the input mode to UI only to stop the user from moving around with the menu open!
- Sets a reference to both the Freelook and First person characters.
- Sets a reference to the Field of View value and the mouse sensitivity value.
- Finally, the base UI widget is hidden from view to stop interactions with it.

Afterwards, inside this event graph you will find events that carry out each of the previously mentioned key functions by calling the respective events from the HUD (**BP_AMHUD**) or by updating values in the game settings or both of the character

settings(**BP_AMFirstPersonCharacter, BP_AMFreeLookPawn**) whenever their respective buttons are clicked (On Clicked events).

Conclusion

The Arcane Mirage Pixel Streaming Template Project provides a solid foundation for creating interactive pixel streaming experiences in Unreal Engine. Whether you need character controllers, mobile controls, or customizable UI elements, this template has you covered. Customize and expand upon these components to develop your unique and engaging pixel streaming projects. Refer to this documentation as a guide for modifying and enhancing the template to meet your project's specific requirements. Enjoy creating your masterpiece!